

[illegible]

RPC  
1-C

[illegible]

```

LL          IIIII
LL          IIIII
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LLLLLLLLLLL IIIII
LLLLLLLLLLL IIIII
SSSSSSSSS
SSSSSSSSS
SS
SS
SS
SS
SSSSSS
SSSSSS
SS
SS
SS
SS
SSSSSSSSS
SSSSSSSSS

```

G 2  
16-Sep-1984 02:17:05  
14-Sep-1984 13:04:23VAX-11 Bliss-32 V4.0-742  
[RPGRTL.SRC]RPGOPEN.B32;1

```
1 0001 0 MODULE RPG$OPEN( XTITLE 'Locate key of reference'
2 0002 0 IDENT = '1-001' ! file:RPGOPEN.B32 EDIT:DJB1001
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1 ++
30 0030 1 FACILITY: RPGII SUPPORT
31 0031 1
32 0032 1 ABSTRACT:
33 0033 1
34 0034 1 This module contains 1 procedure to locate the appropriate key of
35 0035 1 reference in an existing indexed file.
36 0036 1
37 0037 1 ENVIRONMENT: Vax-11 User Mode
38 0038 1
39 0039 1 AUTHOR: D J Braffitt, CREATION DATE: 18-Mar-1983
40 0040 1
41 0041 1 MODIFIED BY:
42 0042 1
43 0043 1 1-001 Original.
44 0044 1 --
45 0045 1
46 0046 1 !<BLF/PAGE>
```

DJB 18-Mar-1983



RPG\$OPEN  
1-001

Locate key of reference  
Declarations

H 2  
16-Sep-1984 02:17:05  
14-Sep-1984 13:04:23

VAX-11 Bliss-32 V4.0-742  
[PPGRTL.SRC]RPGOPEN.B32;1

Page 2  
(2)

```

48 0047 1 %SBTTL 'Declarations'
49 0048 1 |
50 0049 1 | PROLOGUE FILE:
51 0050 1 |
52 0051 1 |
53 0052 1 REQUIRE 'RTLIN:RPGPROLOG';
54 0117 1
55 0118 1
56 0119 1 |
57 0120 1 | LINKAGES
58 0121 1 | NONE
59 0122 1 |
60 0123 1 |
61 0124 1 |
62 0125 1 | TABLE OF CONTENTS:
63 0126 1 |
64 0127 1 |
65 0128 1 FORWARD ROUTINE
66 0129 1 RPG$OPEN_INDEXED;
67 0130 1
68 0131 1 |
69 0132 1 | INCLUDE FILES
70 0133 1 | NONE
71 0134 1 |
72 0135 1 |
73 0136 1 |
74 0137 1 | MACROS
75 0138 1 | NONE
76 0139 1 |
77 0140 1 |
78 0141 1 |
79 0142 1 | EXTERNAL REFERENCES
80 0143 1 |
81 0144 1 |
82 0145 1 EXTERNAL ROUTINE
83 0146 1 RPG$IOEXCEPTION;
84 0147 1
85 0148 1 EXTERNAL LITERAL
86 0149 1 RPG$_ORGNOTIND;

```

| Switches, PSECTs, macros,  
| Linkages and LIBRARYs

```

88 0150 1 %SBTTL 'RPG$OPEN_INDEXED - Locate key of reference'
89 0151 1 GLOBAL ROUTINE RPG$OPEN_INDEXED (
90 0152 1 FAB: REF BLOCK[,BYTE] ! Address of FAB
91 0153 1 )=
92 0154 1
93 0155 1 ++
94 0156 1
95 0157 1 FUNCTIONAL DESCRIPTION:
96 0158 1
97 0159 1 This routine is passed the FAB for an indexed file and opens the file
98 0160 1 based on the appropriate key of reference. The XAB SIZ0, POS0, DTP and NSG
99 0161 1 fields describe the desired key of reference. The goal of this module is to
100 0162 1 get the appropriate key of reference value into the XAB and RAB. Any
101 0163 1 RMS error should be returned as the value of this routine so that
102 0164 1 the generated code will call RPG$IOEXCEPTION. In the case that the file
103 0165 1 organization is not indexed (RMS does not check this, but this module does),
104 0166 1 then RPG$IOEXCEPTION is called directly from this module.
105 0167 1
106 0168 1 The plan of attack is to OPEN or CREATE the file, and if the file exists,
107 0169 1 make sure that it is indexed and then cycle through
108 0170 1 the various key of reference values using $DISPLAY until the
109 0171 1 XAB is set up correctly. Then the RAB key of reference is set
110 0172 1 to the key of reference that has been discovered. The XAB and RAB
111 0173 1 are set initially to the primary key, so if this key matches the
112 0174 1 RPGII programmer's desired key, we are done. The info on the RPGII
113 0175 1 programmer's desired key should be saved before the OPEN for later
114 0176 1 comparison.
115 0177 1
116 0178 1 CALLING SEQUENCE:
117 0179 1
118 0180 1 CALL RPG$OPEN_INDEXED (fab.rr.r)
119 0181 1
120 0182 1 FORMAL PARAMETERS:
121 0183 1
122 0184 1 fab - Address of FAB
123 0185 1
124 0186 1 IMPLICIT INPUTS:
125 0187 1
126 0188 1 NONE
127 0189 1
128 0190 1 IMPLICIT OUTPUTS:
129 0191 1
130 0192 1 NONE
131 0193 1
132 0194 1 ROUTINE VALUE:
133 0195 1
134 0196 1 RMS status value
135 0197 1
136 0198 1 SIDE EFFECTS:
137 0199 1
138 0200 1
139 0201 1 The file pointed to by the fab is opened or created as appropriate.
140 0202 1
141 0203 1 --
```

```
143 0204 2 BEGIN
144 0205 2
145 0206 2 MAP
146 0207 2 FAB : REF BLOCK[, BYTE];
147 0208 2
148 0209 2 LOCAL
149 0210 2 RAB : REF BLOCK[, BYTE];
150 0211 2 XAB : REF BLOCK[, BYTE];
151 0212 2 SAVE_KEY_POS0 : WORD;
152 0213 2 SAVE_KEY_SIZE : BYTE;
153 0214 2 SAVE_KEY_DTP : BYTE;
154 0215 2 RMS_STATUS;
155 0216 2
156 0217 2 RAB = .FAB - RAB$C_BLN;
157 0218 2 XAB = .FAB + FAB$C_BLN + NAM$C_BLN;
158 0219 2 SAVE_KEY_POS0 = .XAB[XAB$W_POS0];
159 0220 2 SAVE_KEY_SIZE = .XAB[XAB$B_SIZE];
160 0221 2 SAVE_KEY_DTP = .XAB[XAB$B_DTP];
161 0222 2
162 0223 2
163 0224 2 + If the CIF (Create IF) bit is set, call $CREATE which will create the
164 0225 2 file if the open fails. The XAB checking should only be done if the
165 0226 2 RMS status is other than RMSS_CREATED.
166 0227 2 If the CIF bit is not set, call $OPEN and then perform XAB checking
167 0228 2 only on success return.
168 0229 2 -
169 0230 2 IF .FAB[FAB$V_CIF]
170 0231 2 THEN
171 0232 2 RMS_STATUS = $CREATE(FAB=.FAB)
172 0233 2 ELSE
173 0234 2 RMS_STATUS = $OPEN(FAB=.FAB);
174 0235 2
175 0236 2 IF .RMS_STATUS AND (.RMS_STATUS NEQ RMSS_CREATED)
176 0237 2 THEN
177 0238 2 BEGIN
178 0239 2 +
179 0240 2 If we get here, the file already exists and has been successfully
180 0241 2 opened. Check that the file organization is indexed.
181 0242 2 -
182 0243 2 IF .FAB[FAB$B_ORG] NEQ FAB$C_IDX
183 0244 2 THEN
184 0245 2 BEGIN
185 0246 2 RAB[RPG$L_CTX_ERROR] = RPG$_ORGNOTIND;
186 0247 2 RPG$IOEXCEPTION(3, .RAB);
187 0248 2
188 0249 2 RETURN RPG$_ORGNOTIND;
189 0250 2 END;
190 0251 2
191 0252 2 +
192 0253 2 If the XAB POS0, SIZE and DTP fields just filled in by the $OPEN match the
193 0254 2 requested POS0, SIZE and DTP fields and the NSG field is 1
194 0255 2 then the current key is to be
195 0256 2 the key of reference and we are done except for setting the RAB KRF
196 0257 2 byte to be the anem as the XAB REF byte; otherwise, we must change the
197 0258 2 key of reference in the XAB and then try a $DISPLAY on the next key
198 0259 2 if there is one.
199 0260 2 -
```



RPG\$OPEN  
1-001

Locate key of reference  
RPG\$OPEN\_INDEXED - Locate key of reference

K 2  
16-Sep-1984 02:17:05  
14-Sep-1984 13:04:23

VAX-11 Bliss-32 V4.0-742  
[RPGRTL.SRC]RPGOPEN.B32;1

Page 5  
(4)

```
200 0261 5      WHILE NOT ((.SAVE_KEY_POS0 EQL .XAB[XAB$W_POS0])
201 0262 5      AND (.SAVE_KEY_SIZE EQL .XAB[XAB$B_SIZE])
202 0263 5      AND (.SAVE_KEY_DTP EQL .XAB[XAB$B_DTP])
203 0264 5      AND (.XAB[XAB$B_NSG] EQL 1)) DO
204 0265 4      BEGIN
205 0266 4      XAB[XAB$B_REF] = .XAB[XAB$B_REF] + 1;
206 0267 4      +
207 0268 4      Invalid key of reference terminates things.
208 0269 4
209 0270 4      Maybe in the future, some more refined messaged could be given
210 0271 4      such as invalid key. In the case of a file that turns out to
211 0272 4      have just one key, a message pointing to the specific attribute
212 0273 4      of that key that is in error from the F spec (position, size, data
213 0274 4      type or number of segments) could be given to the user. For
214 0275 4      now, invalid key of reference is returned if no key matching
215 0276 4      the requested key is found. This is compatible with PDP-11 RPGII.
216 0277 4
217 0278 4      RMS_STATUS = $DISPLAY(FAB=.FAB);
218 0279 4      IF NOT .RMS_STATUS
219 0280 4      THEN
220 0281 4      EX: TLOOP
221 0282 4      END;
222 0283 4
223 0284 4      +
224 0285 4      The RAB key of reference must be set in the case the RMS_STATUS
225 0286 4      is successful.
226 0287 4      -
227 0288 4      IF .RMS_STATUS
228 0289 4      THEN
229 0290 4      RAB[RAB$B_KRF] = .XAB[XAB$B_REF]
230 0291 4      END;
231 0292 4      RETURN .RMS_STATUS
232 0293 4
233 0294 1      END;
```

```
03FC 00000
59 00000000G 8F D0 000C2
54 04 AC D0 00009
53 BC A4 9E 0000D
52 00B0 C4 9E 00011
58 1E A2 B0 00016
57 2E A2 90 0001A
56 13 A2 90 0001E
OB 07 A4 01 E1 00022
00000000G 00 54 DD 00027
01 FB 00029
```

.TITLE RPG\$OPEN Locate key of reference  
.IDENT \1-001\

.EXTRN RPG\$IOEXCEPTION  
.EXTRN RPG\$ ORGNOTIND, SYS\$CREATE  
.EXTRN SYS\$OPEN, SYS\$DISPLAY

.PSECT \_RPG\$CODE, NOWRT, SHR, PIC, 2

```
.ENTRY RPG$OPEN_INDEXED, Save R2,R3,R4,R5,R6,R7,- : 0151
R8,R9
MOVL #RPG$ ORGNOTIND, R9
MOVL FAB, R4 : 0217
MOVAB -68(R4), RAB
MOVAB 176(R4), XAB : 0218
MOVW 30(XAB), SAVE_KEY_POS0 : 0219
MOVB 46(XAB), SAVE_KEY_SIZE : 0220
MOVB 19(XAB), SAVE_KEY_DTP : 0221
BBC #1, 7(R4), 1$ : 0230
PUSHL R4 : 0232
CALLS #1, SYS$CREATE
```

RPG\$OPEN  
1-001

Locate key of reference  
RPG\$OPEN\_INDEXED - Locate key of reference

L 2  
16-Sep-1984 02:17:05  
14-Sep-1984 13:04:23

VAX-11 Bliss-32 V4.0-742  
[RPGRTL.SRC]RPGOPEN.B32;1

Page 6  
(4)

00000000G	00	09	11	00030	BRB	2\$	
	55	54	DD	00032	PUSHL	R4	0234
	54	01	FB	00034	CALLS	#1, SYSS\$OPEN	
00010619	8F	50	DO	0003B	MOVL	R0, RMS_STATUS	0236
		55	E9	0003E	BLBC	RMS_STATUS, 6\$	
		55	D1	00041	CMPL	RMS_STATUS, #67097	
	20	4B	13	00048	BEQL	6\$	0243
		A4	91	0004A	CMPB	29(R4), #32	
F8	A3	13	13	0004E	BEQL	3\$	0246
		59	DO	00050	MOVL	R9, -8(RAB)	0247
		53	DD	00054	PUSHL	RAB	
00000000G	00	03	DD	00056	PUSHL	#3	
	50	02	FB	00058	CALLS	#2, RPG\$IOEXCEPTION	0249
		59	DO	0005F	MOVL	R9, R0	
			04	00062	RET		0261
1E	A2	58	B1	00063	CMPW	SAVE_KEY_POS0, 30(XAB)	
		12	12	00067	BNEQ	4\$	0262
2E	A2	57	91	00069	CMPB	SAVE_KEY_SIZ0, 46(XAB)	
		0C	12	0006D	BNEQ	4\$	0263
13	A2	56	91	0006F	CMPB	SAVE_KEY_DTP, 19(XAB)	
		06	12	00073	BNEQ	4\$	0264
	01	A2	91	00075	CMPB	20(XAB), #1	
		12	13	00079	BEQL	5\$	0266
		17	A2	96	INCB	23(XAB)	0278
		54	DD	0007E	PUSHL	R4	
00000000G	00	01	FB	00080	CALLS	#1, SYSS\$DISPLAY	
	55	50	DO	00087	MOVL	R0, RMS_STATUS	0279
	D6	55	E8	0008A	BLBS	RMS_STATUS, 3\$	
	05	55	E9	0008D	BLBC	RMS_STATUS, 6\$	0287
35	A3	17	A2	90	MOVB	23(XAB), 53(RAB)	0289
	50	55	DO	00095	MOVL	RMS_STATUS, R0	0292
		04	00098	RET			0294

; Routine Size: 153 bytes, Routine Base: \_RPG\$CODE + 0000

: 234 0295 1  
: 235 0296 0 END ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
_RPG\$CODE	153	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded Percent		



RPG\$OPEN  
1-001

Locate key of reference  
RPG\$OPEN\_INDEXED - Locate key of reference

M 2  
16-Sep-1984 02:17:05  
14-Sep-1984 13:04:23

VAX-11 BLISS-32 V4.0-742  
[RPGRTL.SRC]RPGOPEN.B32;1

Page 7  
(4)

```

:
:  $255$DUA28:[SYSLIB]STARLET.L32;1      9776      19      0      581      00:01.0
:  $255$DUA28:[RPGRTL.OBJ]RPGLIB.L32;1    54         1      1      9      00:00.1

```

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS\$:RPGOPEN/OBJ=OBJ\$:RPGOPEN MSRC\$:RPGOPEN/UPDATE=(ENH\$:RPGOPEN)

```














































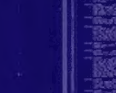













































































































































: Size:          153 code + 0 data bytes
: Run Time:      00:05.6
: Elapsed Time:  00:25.1
: Lines/CPU Min: 3171
: Lexemes/CPU-Min: 13746
: Memory Used:  79 pages
: Compilation Complete

```



0332 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

	RPGMSGTX LIS														
					DTE_DF03 MAP										
RPGMOVE3 LIS															
			RPGSORT LIS												
	RPGOPEN LIS				RTPAD										
					CTDRIVER MAP										
															
															
					RTPAD MAP					RTPADMACS MAR					
RPGMSGPTR LIS															
															
				RPGVECTOR LIS											
	RPGPRINT LIS		RPGUPDATE LIS				RTDEF SDL		DTE_DF03 MAR	CTDRIVER LIS	